УДК 004

Сервер Spring Cloud Config с файловой системой

Семченко Регина Викторовна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Еровлев Павел Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается создание сервера конфигурации. Используется фреймворк Spring Boot. Практическим результатом является рабочий сервер.
Ключевые слова: Spring Boot, Java, сервер

Spring Cloud Config Server with File System

Semchenko Regina Viktorovna
Sholom-Aleichem Priamursky State University
Student

Erovlev Pavel Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article walks you through creating a configuration server. Spring Boot framework is used. The bottom line is a working server.
Keywords: Spring Boot, Java, Server

Spring boot - популярный выбор при разработке микросервисов. Архитектура микросервисов может быть разделена на большое количество небольших сервисов. У каждой службы могут быть собственные конфигурации приложения, помещенные в путь к классу приложения. При увеличении количества служб поддерживать эти свойства конфигурации становится очень сложно. Возможным решением для этого может быть использование общего места для обработки свойств конфигурации. Можно настроить сервер конфигурации, и другие службы могут использовать этот сервер конфигурации для получения необходимых свойств конфигурации.

Цель данной работы - создание сервера конфигурации.

В.И.Зарайский провел обзор на разработку модуля автоматизации работы с конференциями в кафедральном приложении [1]. Р.И.Ибраимов

продемонстрировал процесс создания Docker-образа для Spring Boot проекта и развернул его на платформе AWS EC2[2]. Е.О.Кабардинский, А.Г.Ивашко провели сравнительный анализ сервисных шин предприятия, а так же сравнили некоторые ESB, одна из которых Spring Boot [3]. Так же Р.И.Ибраимов, А.Р.Зайчик, Н.С.Минзатров разработали генеалогическое дерево на языке Java с использованием фреймворка Spring Boot b ,b,kbjntrb gedcom4j[4]. В своей работе А.Б.Джемалетдинов, А.А.Шевченко рассмотрели вопросы создания тестов для Spring Boot mvc контроллеров [5].

Теперь настроим сервер конфигурации, для этого создадим приложение с помощью зависимости «spring-cloud-config-server» (Рис.1).

```
<dependencies>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-config-server</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-dependencies</artifactId>
            <version>${spring-cloud.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```

Рисунок 1 – pom.xml

Добавим следующие свойства конфигурации в файл «application.properties» (Рис.2).

```
server.port=8888
spring.profiles.active=native
spring.cloud.config.server.native.search-locations=classpath:/config/simple-service/
```

Рисунок 2 - application.properties

Поясним код:

- server.port: это порт сервера конфигурации.
- spring.profiles.active: поскольку используется серверная часть файловой системы, необходимо установить для этого свойства значение native .
- spring.cloud.config.server.native.search-locations: это путь к файлу конфигурации (путь к системному файлу, который размещается в любом месте текущей системы).

Поместим файлы конфигурации в каталог, указанный в свойстве «spring.cloud.config.server.native.search-locations». Создадим два файла конфигурации «.yml» с именами «simple-service-dev.yml» и «simple-service-prod.yml».

Эти файлы должны иметь имя с шаблоном «applicationName-profile.yml». В клиентском приложении Spring для загрузки будет задано имя приложения «Spring Simple-service» с двумя активными профилями «dev» и «prod».

Оба файла содержат одно свойство конфигурации «my.prop» с некоторым случайным значением (Рис.3).

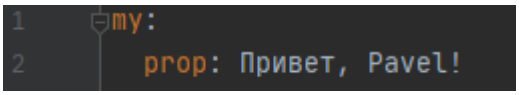

Рисунок 3 - simple-service-dev

Теперь можно проверить, работает ли сервер конфигурации должным образом (рис.4).

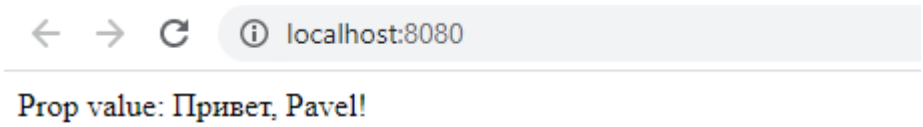

Рисунок 4 – Проверка в браузере

В этой статье было рассмотрено, как настроить сервер облачной конфигурации. Также создана простая служба, используемая в качестве клиента конфигурации для чтения значений свойств.

Библиографический список

1. Зарайский В.И. Разработка модуля автоматизации работы с конференциями в кафедральном приложении // Вестник Ульяновского государственного технического университета. 2019. №3. С. 74-82.
2. Ибраимов Р.И. Развертывание spring приложения с помощью сервиса aws ec2 и docker-контейнеров // Информационно-компьютерные технологии в

экономике, образовании и социальной сфере. 2020. №1(27). С. 138-147.
3. Кабардинский Е.О., Ивашко А.Г. Сравнительный анализ сервисных шин предприятия (esb) // Математическое и информационное моделирование. 2017. №10. С. 177-185.
4. Ибраимов Р.И., Зайчик А.Р., Минзатров Н.С. Разработка генеалогического дерева средствами фреймвока spring boot // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. №4(18). С. 18-23.
5. Джемалетдинов А.Б., Шевченко А.А. Spring boot: создание тестов для spring mvc контроллеров // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. №4(18). С. 104-111.